CSE 462: Project #2 (due 04/26/10)

The additions made on April 12, 2010, are marked in **bold**.

Problem 1 (60 pts)

You will work with an XML database about *PhD ancestry*. For every PhD recipient you need to store:

- the name,
- the university awarding the PhD degree,
- the year of the PhD degree,
- the list of the graduated PhD students.

You can assume that each recipient has exactly one PhD degree and that the names of the PhD recipients are unique. For every university you need to store:

- the name,
- the country.

You can assume that university names are unique.
You will be given an XML document, `phd.xml`, containing the PhD ancestry information.

1. Define the schema of the XML PhD database using DTDs. The schema should reflect the above specification. Avoid creating potential redundancies.
2. The document `phd.xml` should conform to the schema. Validate the document with respect to the schema using one of online XML validation tools.
3. Write the following queries in XQuery:
   (a) *Q1: Find all Witold Lipski's PhD students.*
   (b) *Q2: Find all Alfred Tarski's descendants who graduated from a US university.*
   (c) *Q3: Who are the common ancestors of Krzysztof Apt and Jan Chomicki?*
   (d) *Q4: For every PhD recipient, compute the number of descendants.*
4. Run the queries using Saxon command-line interface and report the results.

Problem 2 (extra credit, 40 pts)

Here, you will work with a relational database representing the same information as the PhD ancestry database in Problem 1. The database will use the interval representation of XML documents. The database will contain **four** tables `Root(Id)`, `Node(Id,Label,Left,Right)`, **Text(Id,TextValue)**, and **Attribute(Id,Attribute,TextValue)**, where `Id` is a unique node identifier, `Label` is the element name, **TextValue is a string**, and `Left` (resp. `Right`) are left (resp. right) interval endpoints. **TextValue in Text represents the text-only content (#PCDATA) of the element node identified by Id**.

1. Write a Java program that traverses the document `phd.xml` using the DOM API and outputs a sequence of SQL INSERT statements to load the PhD ancestry information into the relational database.
2. Run the INSERT statements using ORACLE to populate the relational database.
3. Write the queries *Q1-Q4* in SQL. Run them using ORACLE SQL against the relational database and report the results.

Submission

Submit everything in electronic form using `submit_cse462`. **There will be no deadline extensions.**